REMARKS

Claims 1-40 are pending in the present application. Claims 1-3, 6-21, 23-28 and 30-40 have been rejected. Claims 4, 5 and 22 are objected to. Claim 29 has been canceled. Claims 1, 3, 7-8, 16, 18-21, 23-27, 30-33, 35-36, and 38-39 have been amended. New claims 41-52 have been added. Claims 1-28 and 30-52 remain pending in the application. For the reasons set forth fully below, Applicant respectfully submits that the claims as presented are allowable. Consequently, reconsideration, allowance, and passage to issue are respectfully requested.

Applicant includes a Petition for Extension of Time to extend the deadline for filing a response by one (1) month from December 15, 2003 to January 15, 2004.

35 USC §101 Rejections

The Examiner states:

> **Claim 24 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Descriptive material that cannot exhibit any functional interrelationship with the way in which computing processes are performed does not constitute a statutory process, machine, manufacture or composition of matter and should be rejected under 35 U.S.C. 101. Thus, Office personnel should consider the claimed invention as a whole to determine whether the necessary functional interrelationship is provided. In claim 24 the necessary functional interrelationship is not present, the claimed invention is merely a watermark.**
>
> **Claims 25 and 26 are dependent on rejected claim 35, and is rejected for at least the same reasons.**

Applicant respectfully traverses this rejection.

Claim 24 has been amended to include the functional feature that the watermark is stored, so that it is reproduced when the software is run with a predetermined input sequence. This feature provides a functional interrelationship between the computing processes performed and the chosen watermark. We submit that claims 25 and 26 no longer depend on a rejected claim and therefore the objection to these claims should be withdrawn.

35 USC §112 Rejections

The Examiner states:

Claim 20 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 20 states "the number is derived from a combination of numbers depending on a context and application for the watermark", the claimed feature is has no limitation since it can employ any number under any context.

Applicant respectfully traverses this rejection.

Claim 20 has been amended to specify a class of possibilities, namely deriving the number from a combination of three or more prime numbers.

35 USC §103 Rejections

The Examiner states:

Claims 1-3 and 6-19, 21, 23, 27, 28 and 30-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Moskowitz et al. (US Patent 5,745,569)

As per claim 1,

Moskowitz et al. ('569) discloses a method of watermarking a software object comprising the steps of:

a watermark is stored in the state of the software object. (Column 5, lines 40-51).

Official Notice is taken that "providing an input sequence" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the program would be run with an input sequence because this is necessary for a program to function. The Examiner notes that it is common to all software.

As per claim 1,

Moskowitz et al. ('569) discloses the method as claimed in claim 1

wherein the software object is a program or a piece of a program. (Title)

As per claim 2,

Moskowitz et al. ('569) discloses the method as claimed in claim 1

wherein the software object is a program or a piece of a program. (Title)

As per claim 3,

Moskowitz et al. ('569) discloses the method as claimed in claim 1,

Official Notice is taken that "the state of the software object corresponds to the current values held in a stack, a heap, and global variables of the software object" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the state of the software object may correspond to the current values held in the stack, heap, global variables, registers, program counter of the software because this is necessary for a program to function. The Examiner notes that it is common to all software.

As per claim 6,

Moskowitz et al. ('569) discloses the method of claim 1,

Official Notice is taken that "the watermark is embedded in a topology of a dynamically built graph structure" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the watermark is embedded in the topology of a dynamically built graph structure because this is a fundamental representation of a watermark. The Examiner notes that it is common to steganographic techniques.

As per claim 7,

Moskowitz et al. ('569) discloses the method as claimed in claim 6,

Official Notice is taken that "the dynamically built graph structure corresponds to a representation of a data structure of the program and may be viewed as a set of nodes together with a set of vertices" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the graph structure corresponds to a representation of the data structure of the program and may be viewed as a set of nodes together with a set of vertices because this is a fundamental representation of a program. The Examiner notes that it is common to all software.

As per claim 8,

Moskowitz et al. ('569) discloses the method of claim 1,

further comprising the step of building a recognizer concurrently with the input sequence and the watermark. (Column 6, lines 9-32)

As per claim 9,

Moskowitz et al. ('569) discloses the method of claim 8

wherein the recognizer is a function adapted to identify and extract the watermark from all other dynamic structures on a heap or stack. (Column 6, lines 9-32)

The Examiner notes that as written the term "all other dynamic structures on a heap or stack" comprises the entire program, as it is being run, even if data is read from a hard drive (such as a registration key) it will be stored in an allocated memory position in the heap or the stack.

As per claim 10,

Moskowitz et al. ('569) discloses the method of claim 8

wherein the watermark incorporates a marker that will allow the recognizer to recognize it easily. (Column 6, lines 38-56)

As per claim 10,

Moskowitz et al. ('569) discloses the method of claim 8

wherein the watermark incorporates a marker that will allow the recognizer to recognize it easily. (Column 6, lines 38-56)

As per claim 11,

Moskowitz et al. ('569) discloses the method of claim 8

Official Notice is taken that "the recognizer is retained separately from the program and whereby the recognizer inspects the state of the program" is common and well known in prior art in reference to digital security. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the recognizer is retained separately from the program and whereby the recognizer inspects the state of the program in order to increase the security of the method by separating components of the verification system to make it more difficult to compromise the system. The Examiner notes that it is common for registration programs to exist independently from the programs they register.

As per claim 12,

Moskowitz et al. ('569) discloses the method of claim 8

Official Notice is taken that "wherein the recognizer is dynamically linked with the program when it is checked for the existence of a watermark" is common and well known in prior art in reference to operating systems. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the recognizer is dynamically linked with the program when it is checked for the existence of a watermark in order to utilize memory more efficiently. The Examiner notes that is common in many operating systems to dynamically link and unlink modules (libraries, drivers etc..) from the OS kernel to conserve the amount of memory used by the kernel.

As per claim 13,

Moskowitz et al. ('569) discloses the method of claim 1

Official Notice is taken that "the software object is a part of an application that is obfuscated or incorporates tamper-proofing code" is common and well known in prior art in reference to operating systems. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the application of which the object forms a part is obfuscated or incorporates tamper-proofing code in order to make the code executable. The Examiner notes that obfuscation (i.e. made difficult to read) is common when code is compiled into an executable form.

As per claim 14,

Moskowitz et al. ('569) discloses the method of claim 8,

wherein the recognizer checks the watermark for a signature property. (Column 6, lines 38-56)

As per claim 15,

Moskowitz et al. ('569) discloses the method of claim 14

Official Notice is taken that "the signature property is evaluated by testing for a specific result from a hard computational problem." is common and well known in prior art in reference to digital security. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the signature property is evaluated by testing for a specific result from a hard computational problem in order to make signature non trivial to crack. The Examiner notes that this feature is common to public key encryption (i.e. RSA).

As per claim 16,

Moskowitz et al. ('569) discloses the method of claim 14

including the step of creating a number having at least one numeric property which is embedded in the topology of the watermark whereby the signature property is evaluated by testing the at least one or more numeric property. (Column 6, lines 38-56)

As per claim 17,

Moskowitz et al. ('569) discloses the method of claim 16

Official Notice is taken that "the signature property is evaluated by testing whether the number is a product of two primes" is common and well known in prior art in reference to digital security. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the signature property is evaluated by testing whether n is the product of two primes order to make signature non trivial to crack. The Examiner notes that this feature is common to public key encryption (i.e. RSA).

As per claim 18,

Moskowitz et al. ('569) discloses a method of verifying the integrity or origin of a program including the steps of:

watermarking the program with a watermark, wherein the watermark is stored in the state of a program as the program is being run (Column 5, lines 40-51)

Official Notice is taken that "being run with a input sequence" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the program would be run in a sequence because this is necessary for a program to function. The Examiner notes that it is common to all software.

building a recognizer concurrently with the input and watermark wherein the recognizer is adapted to extract the watermark from other dynamically allocated data wherein the recognizer is adapted to check for a number n.(Column 6, lines 38-56)

Official Notice is taken that "the recognizer is retained separately from the program and whereby the recognizer inspects the state of the program" is common and well known in prior art in reference to digital security. It would have been obvious to one having ordinary skill in the art at the time the invention was made that R is retained separately from the program and whereby R inspects the state of the program in order to increase the security of the method by separating components of the verification system to make it more difficult to compromise the system. The

Examiner notes that it is common for registration programs to exist independently from the programs they register.

As per claim 19

Moskowitz et al. ('569) discloses the method of claim 18

Official Notice is taken that "the number is the product of two primes and wherein the number is embedded in the topology of the watermark" is common and well known in prior art in reference to digital security. It would have been obvious to one having ordinary skill in the art at the time the invention was made that n is the product of two primes and wherein n is embedded in the topology of W in order to make signature non trivial to crack. The Examiner notes that this feature is common to public key encryption (i.e. RSA).

As per claim 21,

Moskowitz et al. ('569) discloses the method of claim 18

Official Notice is taken that "adapted to be resistant to tampering, the resistance to tampering capable of being by means of obfuscation or by adding tamper-proofing code" is common and well known in prior art in reference to operating systems. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the application of which the object forms a part is obfuscated or incorporates tamper-proofing code in order to make the code executable. The Examiner notes that obfuscation (i.e. made difficult to read) is common when code is compiled into an executable form.

As per claim 23,

Moskowitz et al. ('569) A method of watermarking software.

Official Notice is taken that "embedding a watermark in a static string; and applying an obfuscation technique whereby this static string is converted into executable code" is common and well known in prior art in reference to operating systems. It would have been obvious to one having ordinary skill in the art at the time the invention was made to embed a watermark in a static string; and apply an obfuscation technique whereby this static string is converted into executable codein order to make the code executable. The Examiner notes that obfuscation (i.e. made difficult to read) is common when code is compiled into an executable form.

As per claim 27

Moskowitz et al. ('569) discloses a method of fingerprinting software comprising the steps of:

providing a plurality of watermarked programs, the plurality of watermarked programs being obtained by providing an input sequence for each program of the plurality of programs and storing a watermark in a state of a software object for the program. (Column 5, lines 40-51)

Official Notice is taken that "being run with a input sequence" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the program would be run in a sequence because this is necessary for a program to function. The Examiner notes that is common to all software.

As per claim 28,

Moskowitz et al. ('569) discloses the method of fingerprinting software as claimed in claim 27.

Official Notice is taken that "the plurality of watermarked programs each of which has a number with a common prime factor" is common and well known in prior art in reference to digital security. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the plurality of watermarked programs each of which has a number with a common prime factor in order to make signature non trivial to crack. The Examiner notes that this feature is common to public key encryption (i.e. RSA).

As per claim 30,

Moskowitz et al. ('569) discloses a computer readable medium including a program for watermarking a software object, the program including instructions for:

storing a watermark in the state of the software object as the software object is being run with the input sequence. (Column 5, lines 40-51)

Official Notice is taken that "providing an input sequence" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the program would be run with an input sequence because this is necessary for a program to function. The Examiner notes that it is common to all software.

As per claim 31,

Moskowitz et al. ('569) discloses a computer comprising:

a software object, an input sequence; a watermark stored in the state of the software object as the software object is being run with the input sequence.(Column 5, lines 40-51)

As per claim 32,

Moskowitz et al. ('569) discloses a method of fingerprinting software comprising the steps of:

providing a plurality of watermarked programs, the plurality of watermarked programs being obtained by watermarking each program of the plurality of programs with a watermark. (Abstract)

wherein the watermark is stored in the state of a program as the program is being run with an input sequence (Column 5, lines 40-51)

building a recognizer concurrently with the input sequence and watermark W wherein the recognizer is adapted to extract the watermark from other dynamically allocated data (Column 6, lines 9-32)

Official Notice is taken that "the recognizer is retained separately from the program and whereby the recognizer inspects the state of the program" is common and well known in prior art in reference to digital security. It would have been obvious to one having ordinary skill in the art at the time the invention was made that R is retained separately from the program and whereby R inspects the state of the program in order to increase the security of the method by separating components of the verification system to make it more difficult to compromise the system. The Examiner notes that it is common for registration programs to exist independently from the programs they register.

As per claim 33,

Moskowitz et al. ('569) discloses a method of fingerprinting software comprising the steps of:

providing a plurality of watermarked programs, the plurality of watermarked programs being obtained by watermarking each program of the plurality of programs with a watermark, (Abstract)

Official Notice is taken that "the watermark being obtained by embedding a watermark in a static string and applying an obfuscation technique whereby the static string is converted into executable code" is common and well known in prior art in reference to operating systems. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the application of which the object forms a part is obfuscated or incorporates tamper-proofing code in order to make the code secure. The Examiner notes that obfuscation (i.e. made difficult to read) is common when code is compiled into an executable form.

A per claim 34,

Moskowitz et al. ('569) discloses a method of fingerprinting software comprising the steps of:

providing a plurality of watermarked programs, the plurality of watermarked programs being obtained by watermarking each program of the plurality of programs with a watermark, (Abstract)

Official Notice is taken that "the watermark being obtained by choosing a watermark from a class of graphs having a plurality of members, each member of the class of graphs has at least one property, the at least one property being capable of being tested by integrity-testing software and applying the watermark to the software" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the watermark is chosen from a class of graphs because this is a fundamental representation of a watermark. The Examiner notes that it is common to steganographic techniques.

As per claim 35,

Moskowitz et al. ('569) discloses a computer-readable medium including a program for verifying the integrity or origin of a program, the program including instructions for:

watermarking the program with a watermark, wherein the watermark is stored in the state of a program as the program is being run with an input sequence; (Column 5, lines 40-51)

building a recognizer concurrently with the input sequence; (Column 5, lines 40-51)

building a recognizer concurrently with the input sequence and watermark wherein the recognizer is adapted to extract the watermark from other dynamically allocated data (Column 6, lines 9-32)

Official Notice is taken that "the recognizer is retained separately from the program and whereby the recognizer inspects the state of the program" is common and well known in prior art in reference to digital security. It would have been obvious to one having ordinary skill in the art at the time the invention was made that R is retained separately from the program and whereby R inspects the state of the program in order to increase the security of the method by separating components of the verification system to make it more difficult to compromise the system. The Examiner notes that it is common for registration programs to exist independently from the programs they register.

As per claim 36,

Moskowitz et al. ('569) discloses a computer-readable medium including a program for watermarking software, the program including instructions for:

embedding a watermark in a static string; (Abstract)

Official Notice is taken that "applying an obfuscation technique whereby the static string is converted into executable code" is common and well known in prior art in reference to operating systems. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the application of which the object forms a part is obfuscated or incorporates tamper-proofing code in order to make the code secure. The Examiner notes that obfuscation (i.e. made difficult to read) is common when code is compiled into an executable form.

As per claim 37,

Moskowitz et al. ('569) discloses a computer-readable medium including a program for watermarking software, the program including instructions for:

applying the watermark to the software, (Abstract)

Official Notice is taken that "the watermark being obtained by choosing a watermark from a class of graphs having a plurality of members, each member of the class of graphs has at least one property, the at least one property being capable of being tested by integrity-testing software" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the watermark is chosen from a class of graphs because this is a fundamental representation of a watermark. The Examiner notes that it is common to steganographic techniques.

As per claim 38,

Moskowitz et al. ('569) discloses a computer capable of verifying the integrity or origin of a program, the computer comprising:

an input sequence; a watermark for watermarking the program, wherein the watermark is stored in the state of a program as the program is being run with the input sequence; (Column 5, lines 40-51)

a recognizer built concurrently with the input sequence and watermark wherein the recognizer is adapted to extract the watermark from other dynamically allocated data (Column 6, lines 9-32).

Official Notice is taken that "the recognizer is retained separately from the program and whereby the recognizer is adapted to check for a number" is common and well known in prior art in reference to digital security. It would have been obvious to one having ordinary skill in the art at the time the invention was made that R is retained separately from the program and whereby R inspects the state of the program in order to increase the security of the method by separating components of the verification system to make it more difficult to compromise the system. The Examiner notes that it is common for registration programs to exist independently from the programs they register.

As per claim 39,
Moskowitz et al. ('569) discloses a computer for watermarking software comprising:
a static string; a watermark embedded in the static string;(Abstract)
Official Notice is taken that "an obfuscation technique whereby the static string is converted into executable code" is common and well known in prior art in reference to operating systems. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the application of which the object forms a part is obfuscated or incorporates tamper-proofing code in order to make the code secure. The Examiner notes that obfuscation (i.e. made difficult to read) is common when code is compiled into an executable form.
As per claim 40,
Moskowitz et al. ('569) discloses a computer comprising:
software to which the watermark is applied. (Abstract)
Official Notice is taken that "the watermark being obtained by choosing a watermark from a class of graphs having a plurality of members, each member of the class of graphs has at least one property, the at least one property being capable of being tested by integrity-testing software" is common and well known in prior art in reference to computer programs. It would have been obvious to one having ordinary skill in the art at the time the invention was made that the watermark is chosen from a class of graphs because this is a fundamental representation of a watermark. The Examiner notes that it is common to steganographic techniques.

Applicant respectfully traverses this rejection. Moskowitz teaches a method for protecting copy data protection code from unauthorized modifications. This is in contrast to the claimed invention, which relates to embedding digital watermarks in software.

Furthermore, referring in particular to claim 1, this claim requires that the watermark is stored in a state of the software object. The state of the software object, being dynamic, is not apparent to the user. Moskowitz teaches the formation of a static watermark that is detectable in the program object in its originally-distributed form. Claims 2, 3, 6-17 are submitted to be novel and inventive at least for the same reasons due to depending on claim 1. Claim 3 has been amended to clarify that the watermark is detected by detecting aspects of the state of the software object.

It is submitted that claim 18 is novel and inventive for at least the same reason as claim 1 and that claims 19 and 21 are novel and inventive at least due to depending on claim 18. Claim 21 has been amended to clarify that normal compilation of code into executable form is excluded from its scope by specifying that the obfuscation is applied to a program (object code) that is already compiled, or to a program (source code) that has not yet been compiled.

Claim 23 has been amended and now requires that the string be reproduced when at least one input is applied to the software. This is to be contrasted with normal compilation, where a watermark string is not able to be reproduced. Claims 33, 36 and 39 are submitted to be novel and inventive for the same reasons.

Claims 27, 30, 31 and 32 have been amended to clarify that the watermark is stored in a state of a software object for the program, with the watermark being detectable as the software object is being run with a particular input sequence. The watermark is therefore dynamically built. This is in contrast to Moskowitz, which teaches the use of a key to access certain parts of the executable code and obfuscating the essential code resources of the product into data resources, making it less recognizable. Claim 28 is submitted to be inventive at least for depending on claim 27.

It is submitted that claim 34 is inventive. The Applicant submits that there is no prior art that teaches the use of a graph as a representation of a watermark. The Examiner says that representing watermarks as graphs is common to steganographic techniques. The Applicant respectfully submits that in the great majority of cases in steganographic techniques, the watermark is represented as a numeric "signal" that is steganographically written onto a numeric "cover message", where the cover message is either unspecified or is a digitally sampled photograph or sound signal. Claims 37 and 40 are submitted to be novel and inventive for th same reasons.

Claim 35 has been amended to clarify that the watermark becomes detectable when the program is being run with the input sequence. It is submitted that this claim is novel and inventive for at least the same reasons as claims 1 and 30.

Furthermore, the Examiner says that retaining the recognizer separately from the program and whereby the recognizer inspects the state of the program is common and well known in prior art in reference to digital security. Even if this were the case, the Applicant submits that it would not be obvious to combine this information with the teachings of Moskowitz et al., because the technique described in Moskowitz et al. makes it essential that the recognizer (which extracts the executable code from the data resource) forms part of the program (see column 6, lines 29-34 of Moskowitz et al.).

The Applicant submits that the combination of a watermark that is stored in the state of a program so that the watermark becomes detectable when the program is being run with an input sequence with a separate recognizer has particular advantages and is novel and inventive. In particular, this combination provides a stealthy and resilient watermark. The only dynamic watermarks known prior to the priority date of the claims were Easter Eggs, where the watermark is not stored in the dynamically allocated data. Claim 38 is submitted to be novel and inventive for the same reasons.

New Claims 41-49

New claims 41-49 are added to further define the scope and novelty of the present invention and are allowable for at least the reasons set forth for claims 1-28 and 30-40.

Allowable Subject Matter

Claims 4, 5 and 22 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and nay intervening claims and complying with double patenting statutes.

Applicant appreciates and acknowledges Examiner's indication of allowability of claims 4, 5 and 20. New claims 50-52 have been added. Independent claim 50 is claim 4 rewritten in independent form. Independent claim 51 is claim 5 rewritten in independent form. Independent claim 52 is claim 22 rewritten in independent form. Applicant submits therefore, that claims 50-52 are allowable over the cited references.

Conclusion

Accordingly, Applicant respectfully requests reconsideration and allowance of claims 1-28 and 30-52 as now presented.

Applicant's attorney believes that this application is in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicant's attorney at the telephone number indicated below.

Respectfully submitted,
SAWYER LAW GROUP LLP

January 15, 2004
Date

Stephen G. Sullivan
Attorney for Applicant(s)
Reg. No. 38,329
(650) 493-4540